


When Designs of Computer Architecture Meet Machine Learning

Tsung Tai Yeh

Computer Science Department of
National Chiao Tung University,
Taiwan

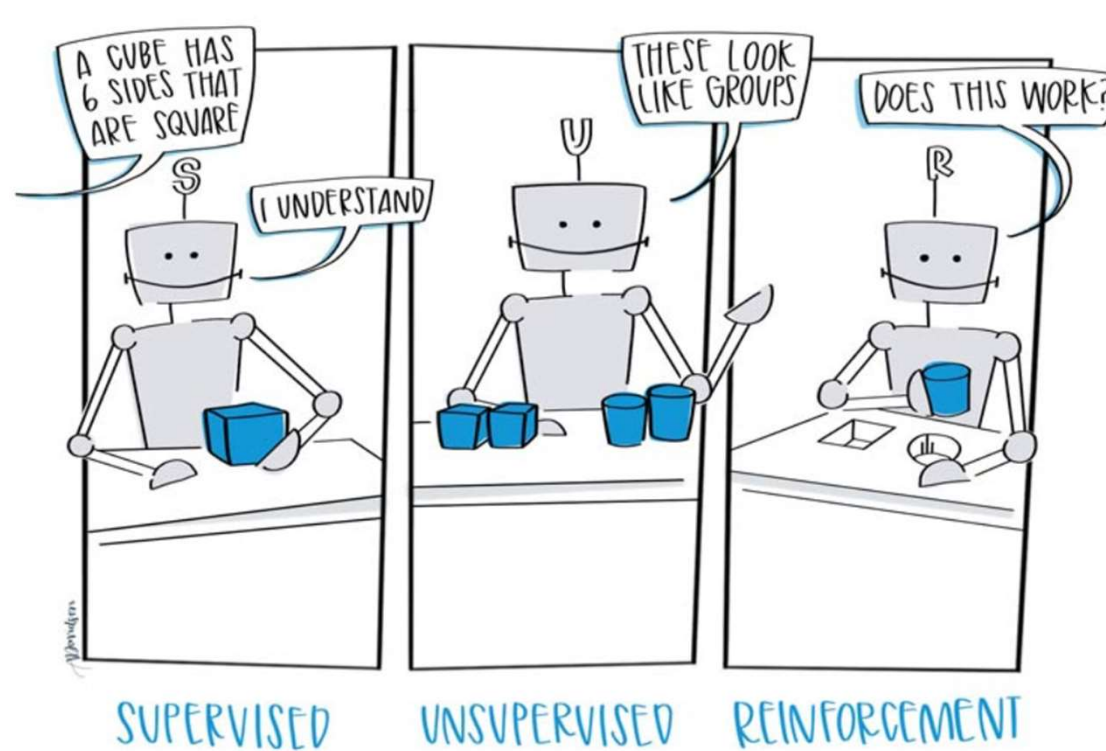
Overview

- Machine Learning & Deep Neural Network
- Golden Age of Microprocessor Design
- Domain Specific Accelerator
- My Research and my CAS Lab at NCTU
- Life @ the U.S.
- Advice for students

Machine Learning & Deep Neural Network

What is Machine Learning ?

- “Giving computers the ability to learn without being explicitly programmed” – Arthur Samuel, 1959s



<https://www.ceralytics.com/3-types-of-machine-learning/>

Supervised Learning

- **Data:** (x, y)

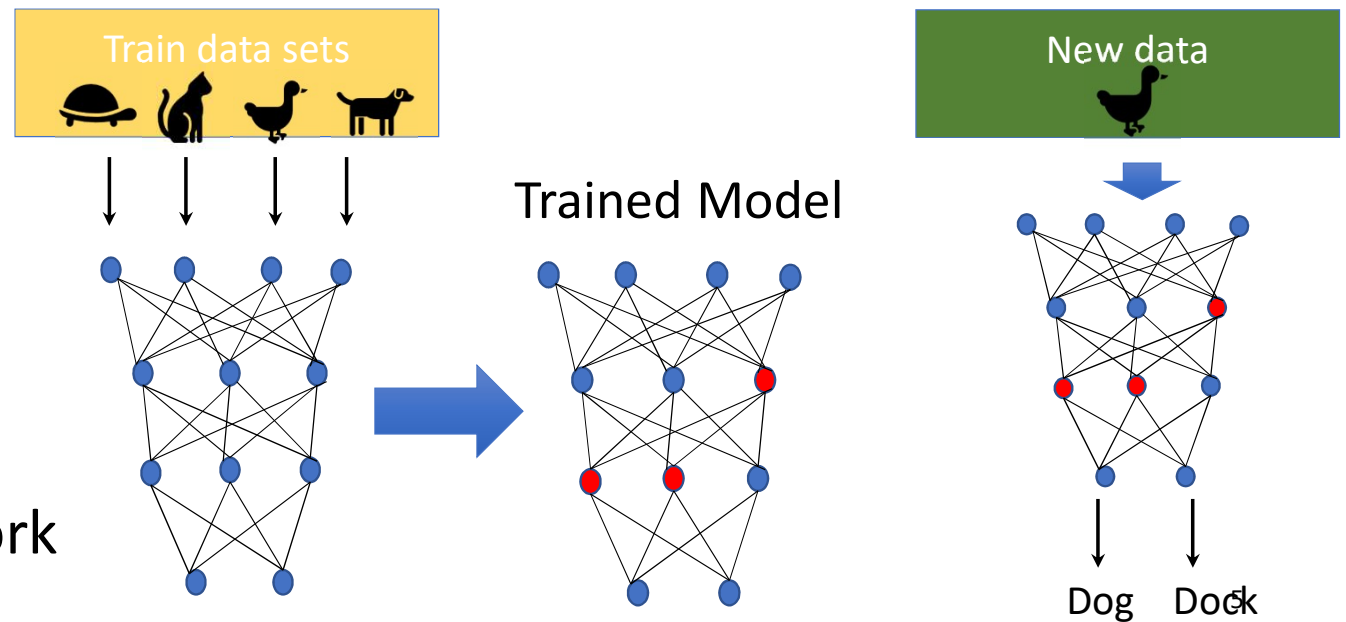
The x is data, y is label

- **Goal:** Learn a function to map x from label data y

- **Examples:** Object detection, classification, image captioning etc..

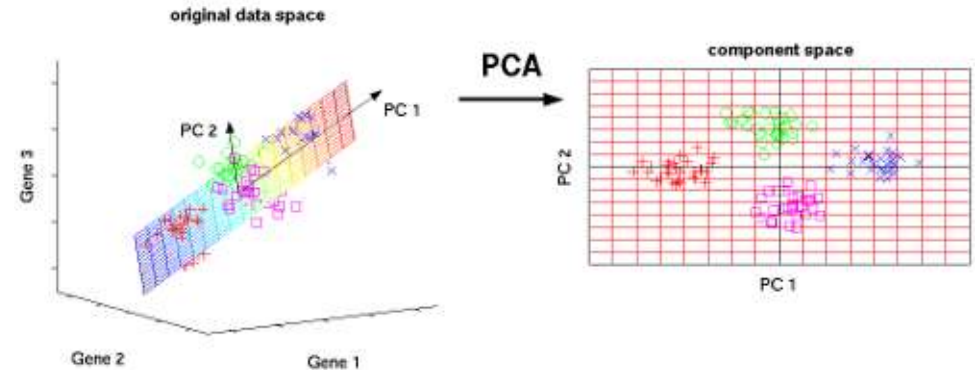
- **Problems:**

- Tedious labelling work



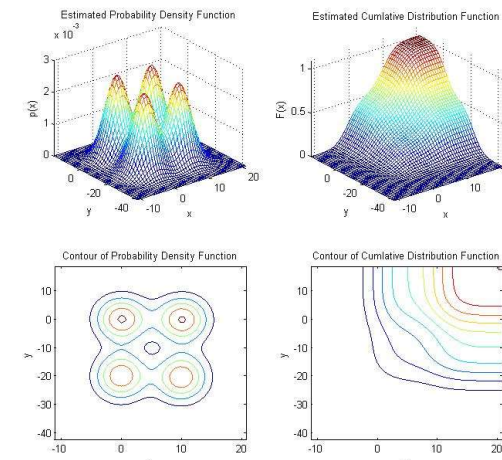
Unsupervised Learning

- **Data:** x , no labels !
- **Goal:** Learn underlying hidden structures of the data
- **Example:** Clustering, feature learning, density estimation, dimensionality reduction etc..
- **Problem:**
 - The curse of dimensionality



Dimension Reduction on PCA

http://www.nl pca.org/pca_principal_component_analysis.html

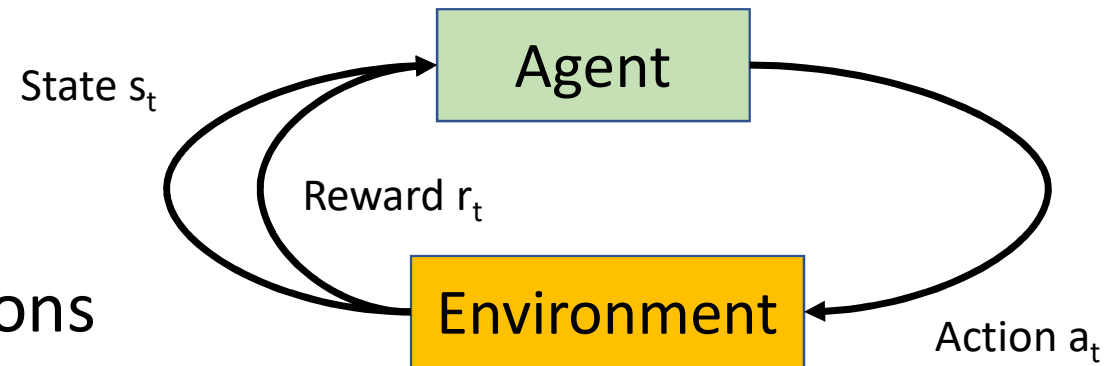


2-D Density Estimation

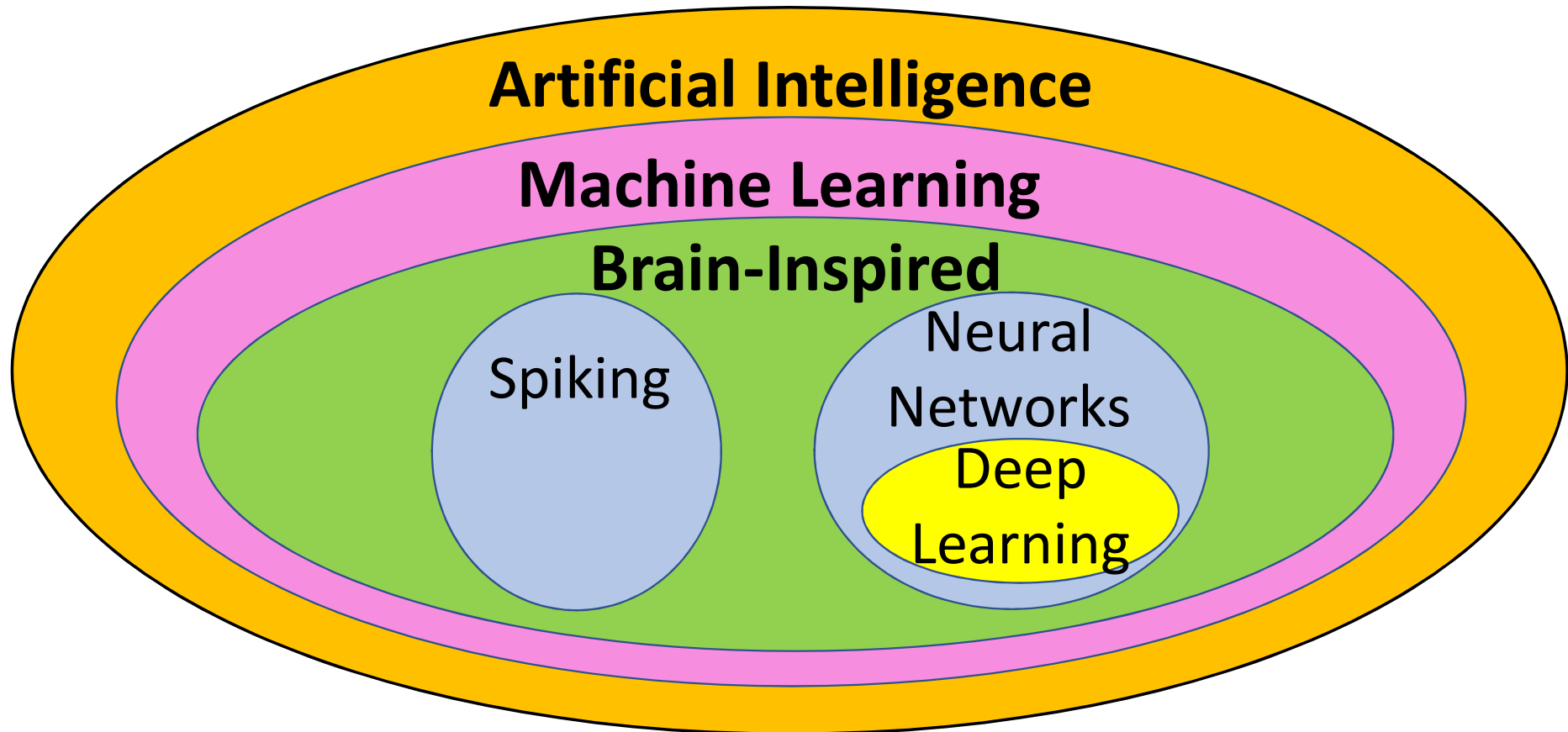
<https://www.mathworks.com/matlabcentral/fileexchange/19280-bivariate-kernel-density-estimation-v2-1>

Reinforcement Learning

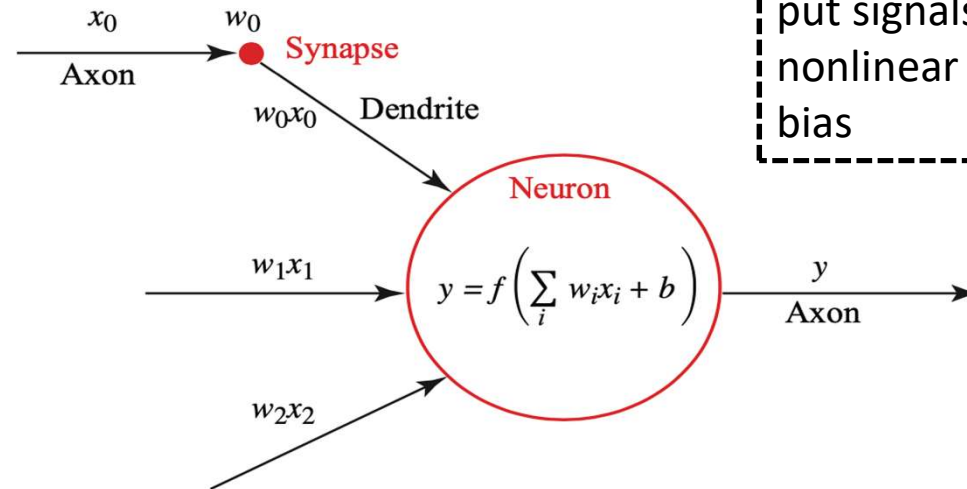
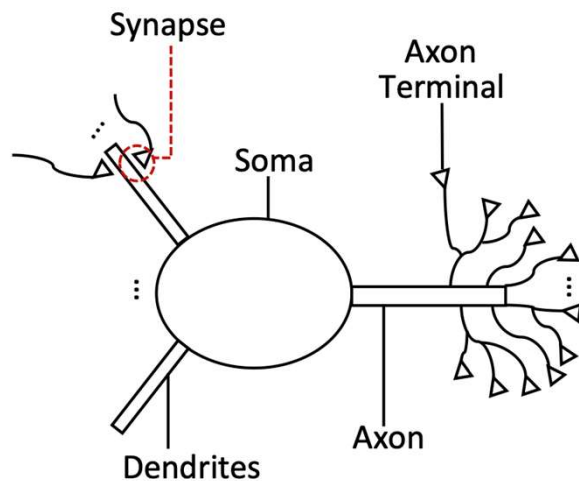
- An **agent** interacts with the **environment**
- Learning from the **reward** signals
- **Goal:** Learn how to take actions to maximize the reward
- **Examples:** Robots control, Deep mind AlphaGo, Atari Gaming
- **Problems:**
 - Reliability



Deep Learning



How does the brain work ?

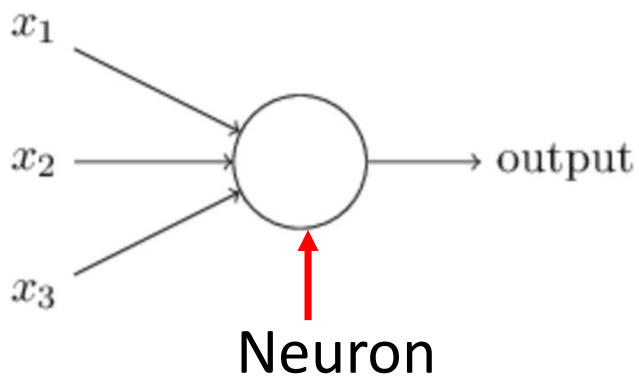


$x, w, f(), b$ are activations(input/output signals), weights, nonlinear function, bias

- **Neurons** (86 B) (perception) are assembled into layers which are connected via synapses
- **Dendrites** receive inputs from upstream neurons via the synapses.
- Soma membrane fires inputs to an axon.
- **Axons** terminals transmits outputs to downstream neurons.

How does neuron (perception) work?

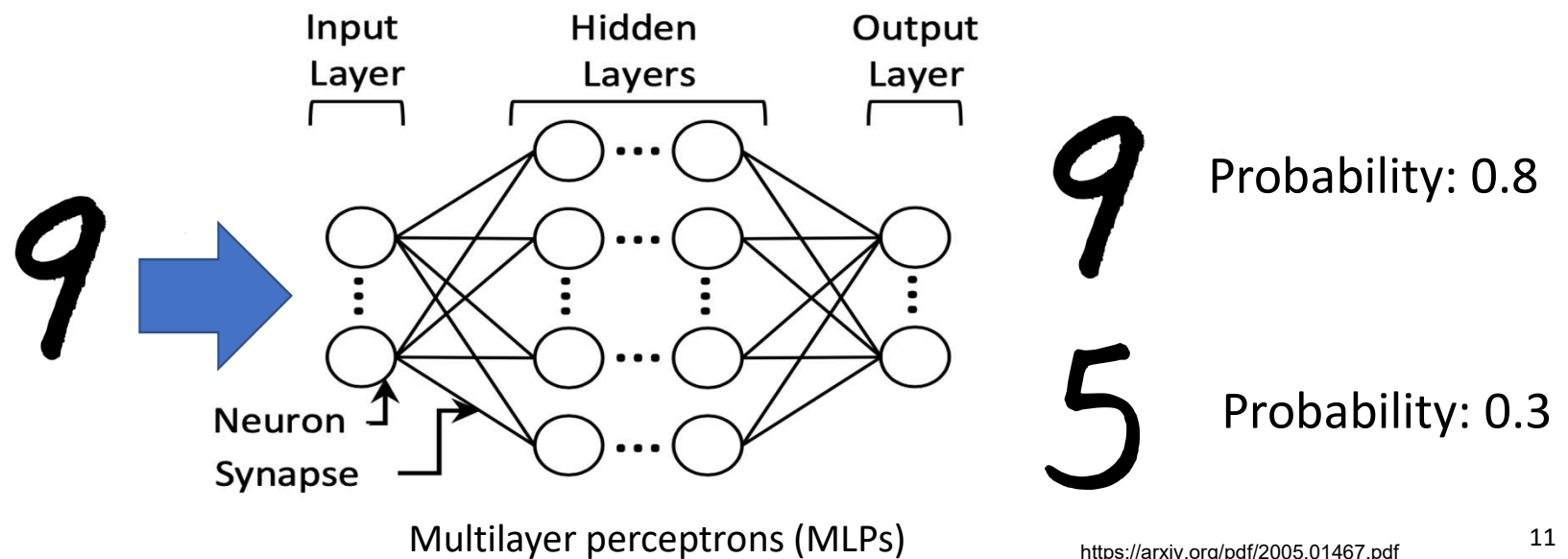
- Frank Rosenblatt proposed this neural model in 1950-1960
- One **neuron** can have multiple inputs (x_i)
- **Weight** expresses the importance of the respective inputs to the output
- The output is determined by the rule with **weighted sum/threshold**
- Neuron is a device that makes decisions by weighting up evidence



$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

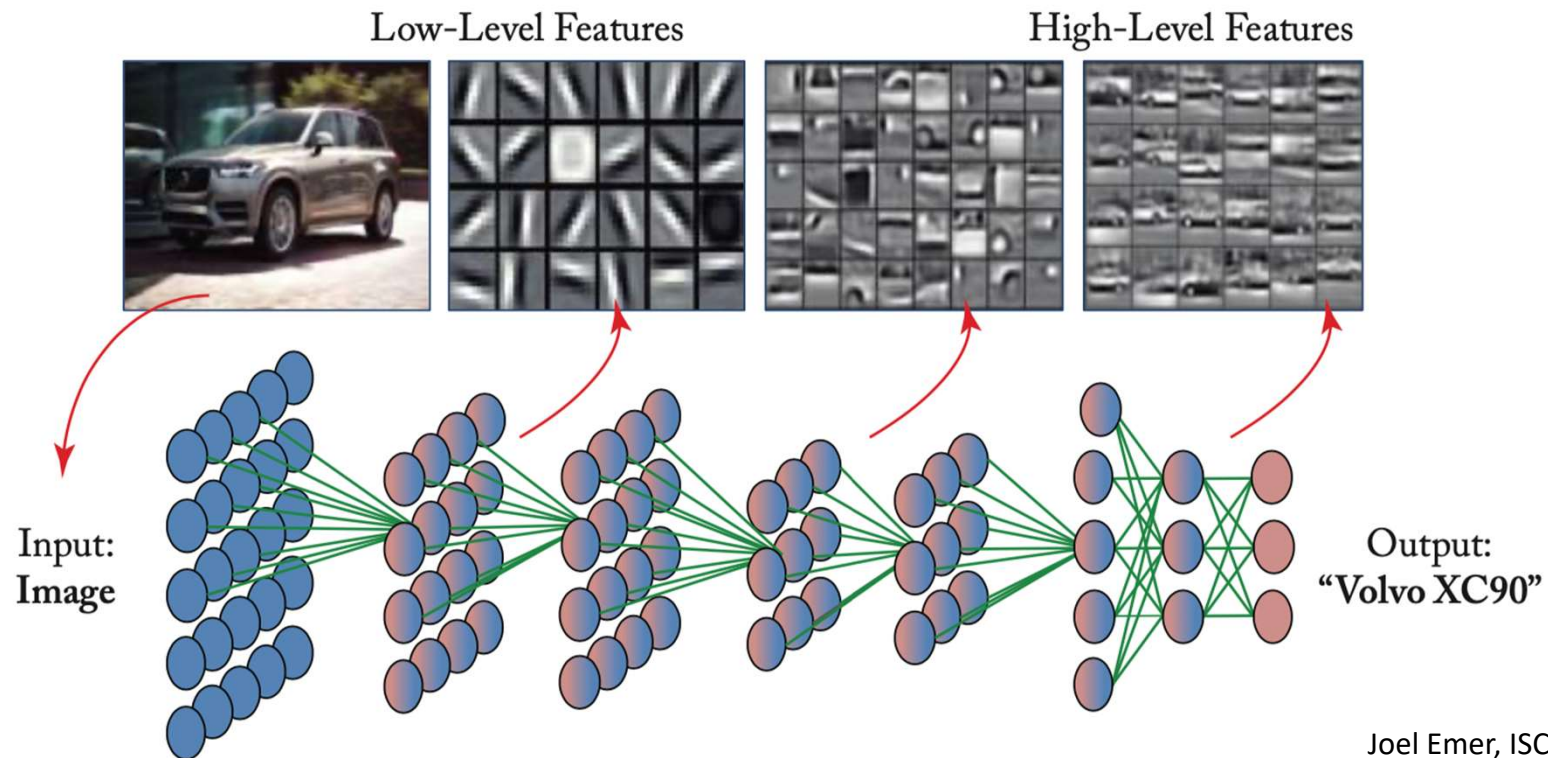
Neural Networks (NN)

- **Hidden layer:** neurons in this layer are neither inputs nor outputs, extracting input features
- To encode the **intensities** of image pixels to the input neurons
- Picking the output values > 0.5 that indicates input image is 9



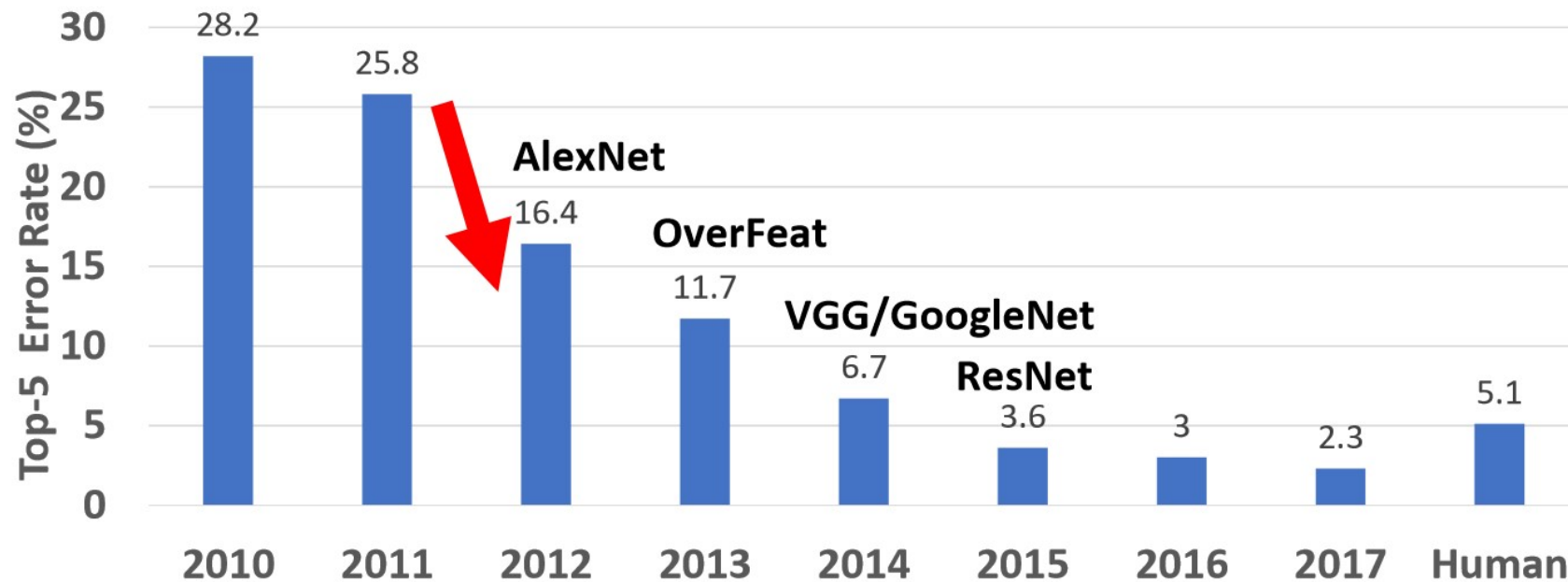
What is Deep Learning ?

- DNN has more than **3** layers (more than one hidden layer)
- DNNs can learn high-level features than shallow neural networks



Why Deep Neural Network become popular?

- DNN model outperforms human-being on the ImageNet Challenge



<https://arxiv.org/ftp/arxiv/papers/1911/1911.05289.pdf>

Convolutional Neural Networks

Deep CNN: 5 – 1000 Layers

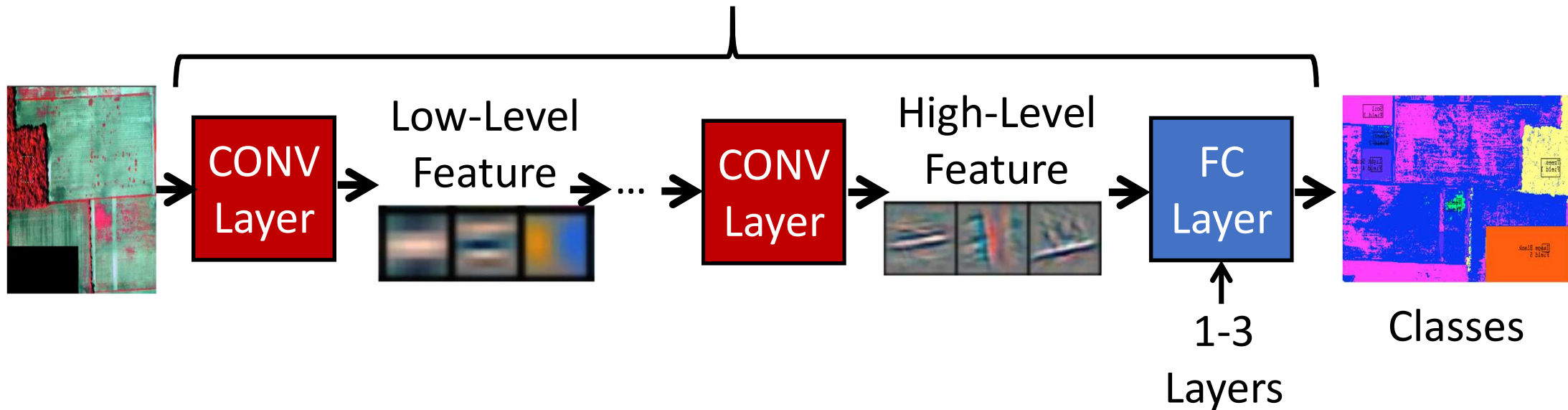
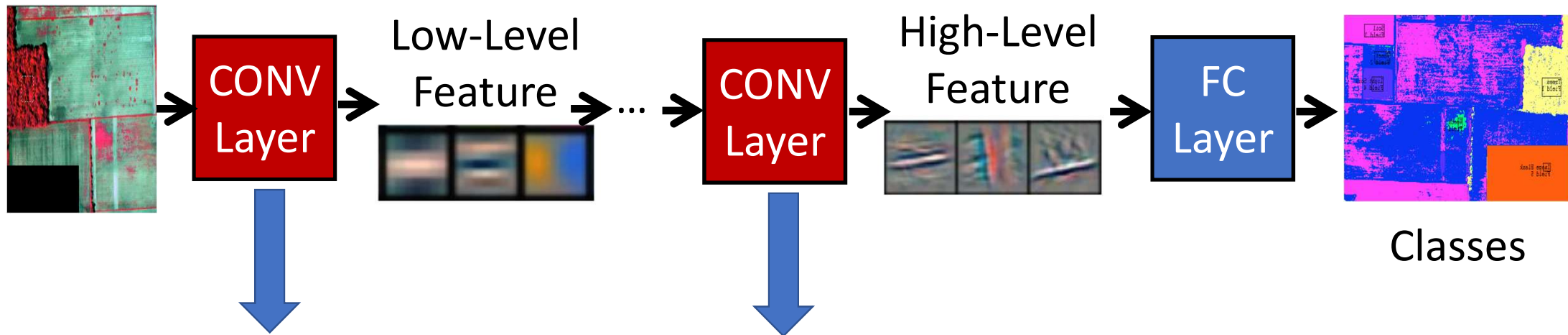


Image classification Pipeline:

Input -> Processing in Deep Layers + Trained Weights -> Output (with Classes)

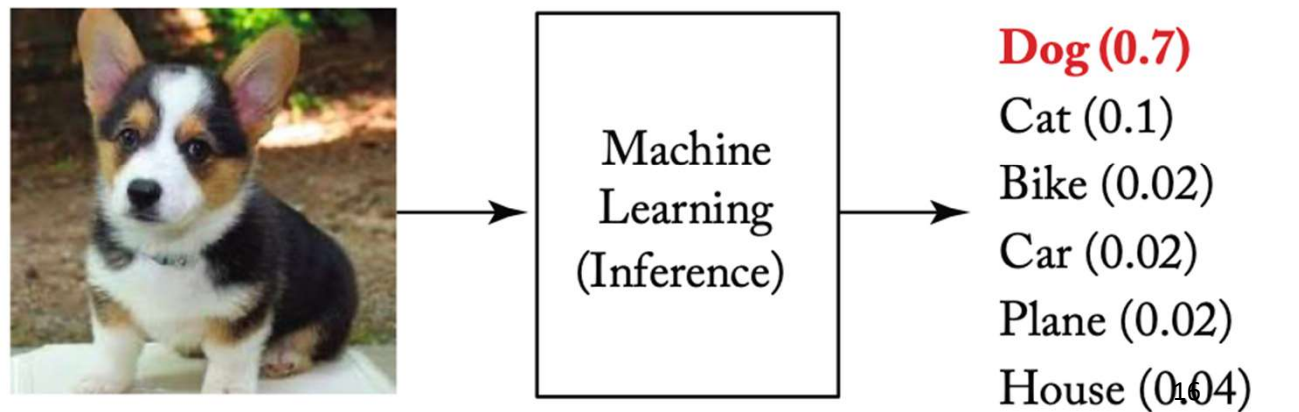
Convolutional (CONV) Layers



1. Convolutions mainly perform **vector-and-matrix multiplication**.
2. Convolutions takes more than **90%** of overall computation (critical path).
3. Optimization (software/hardware) for convolutions matters.

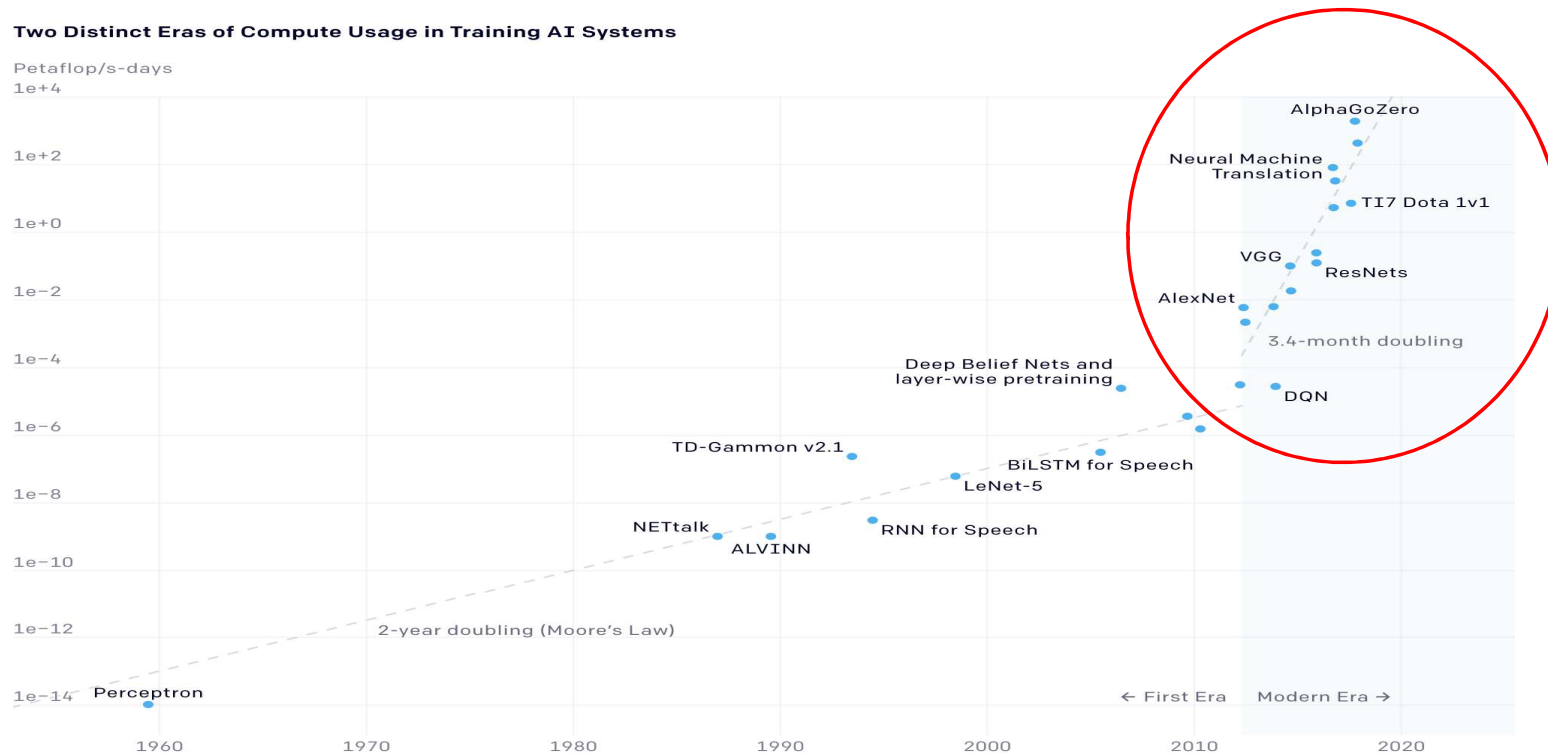
Training versus Inference

- **Training:** Determining the value of the **weights** in the network
 - Minimizing loss (L)
 - Loss (L): the gap between ideal correct probabilities and the probabilities computed by the DNN model
- **Inference:** Apply trained weights to determine output
Include only forward



No free lunch on DNN computation

- AlexNet to AlphaGo Zero: A 300,000 x Increase in Compute



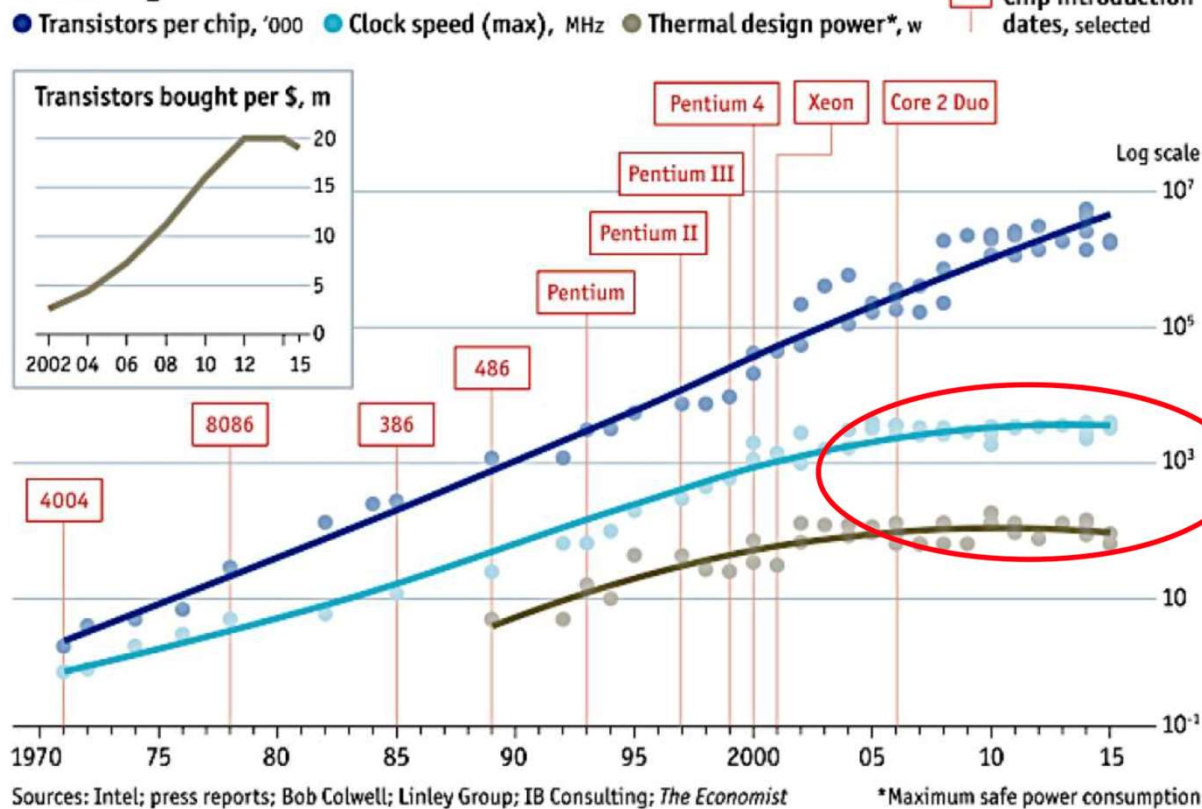
Speed up Machine Learning through Domain-Specific Accelerator

A Golden Age in Microprocessor Design

- A great leap in microprocessor speed $\sim 10^6$ X faster over 40 years
- Architectural innovations
 - Width: 8- \rightarrow 16- \rightarrow 32- \rightarrow 64 bits (~ 8 X)
 - Instruction level parallelism (ILP)
 - Multicore: 1 processor to 16 cores
 - Clock rate: 3 – 4000 MHz (~ 1000 X through technology & architecture)
- IC technology makes it possible
 - **Moore's Law**: growth in transistor count (2X every 1.5 years)
 - **Dennard Scaling**: power/transistor shrinks at the same rate as transistors are added

Increasing transistors is not getting efficient

Stuttering



General purpose processor is not getting faster and power-efficient because of

Slowdown of Moore's Law and Dennard Scaling

Need **Specialized/Domain-specific accelerators** to improve computing speed and energy

Moore's Law

- The number of transistors per chip **doubles** every 18-24 months
- That has not been true for years
- It is getting to be increasingly difficult to maintain this exponential improvement !! Why?

Dennard Scaling

- As the size of the transistor becomes **small**
 - The voltage is reduced
 - Circuits can be operated at higher frequency at the same power

Related to
transistor
size



$$\mathbf{Power} = \text{alpha} \times \mathbf{C} \mathbf{F} \mathbf{V}^2$$

alpha: percent time switched

C: capacitance

F: Frequency

V: Voltage

What's wrong on
Dennard Scaling?

Dennard Scaling ignores "leakage current" , "threshold voltage"

So, as transistors get small, power density increases !!

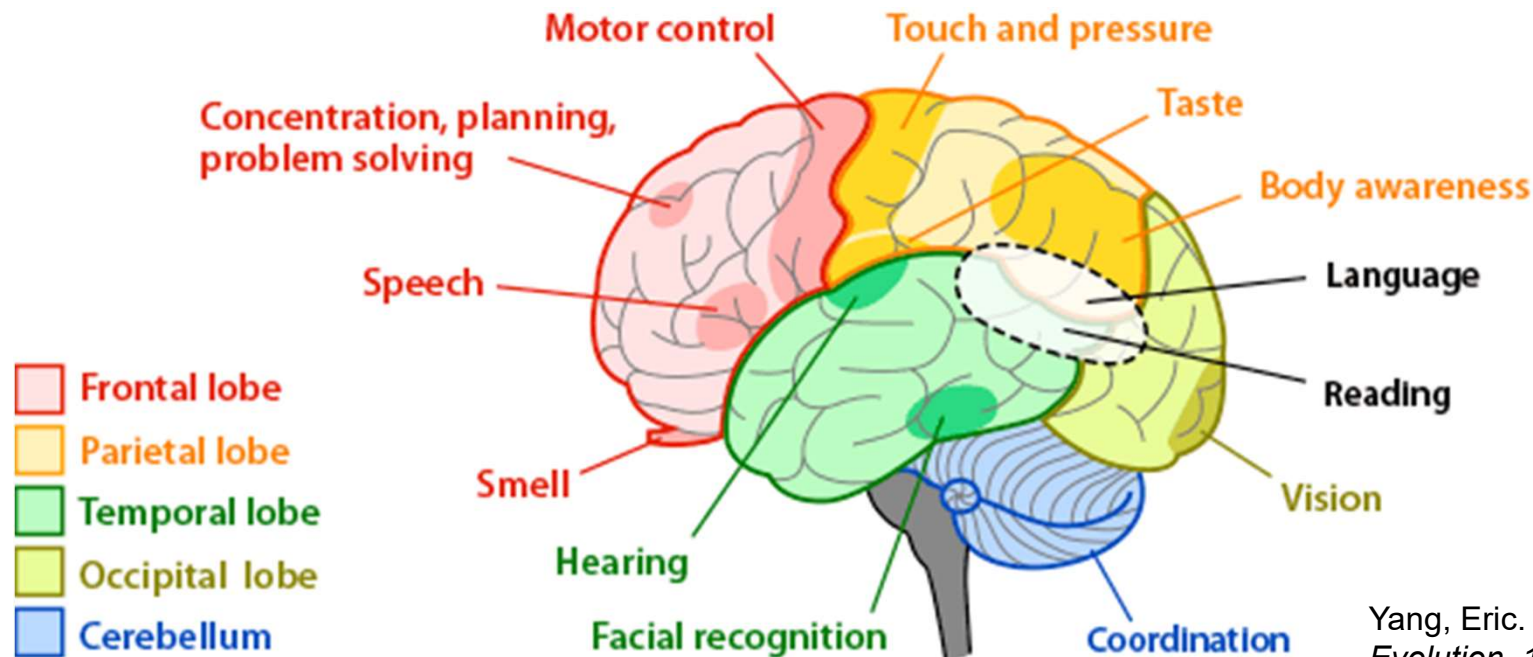
What's Left ?

- Transistors not getting much better
- Power budget not getting much higher
- One inefficient processor/chip to N efficient processors/chip
- Only path left is **Domain Specific Architectures**
 - Just do a few tasks, but extremely well

Uncover Your Brain

2400 kcal/24 hr = 100 kcal/hr = 27.8 cal/
sec = 116.38 J/s = 116 W
20% x 116 W = 23.3 W

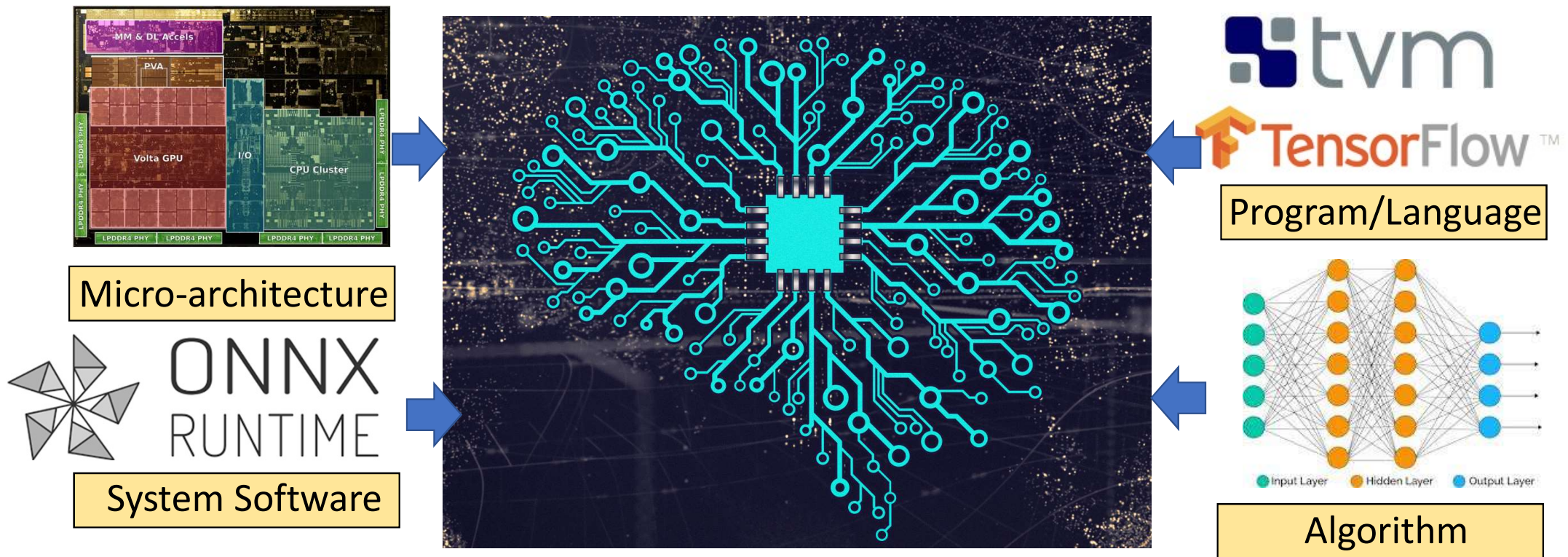
- The human-being brain comprises different areas (accelerators)
- An adult brain only consumes about 23 W a day !! (Yang)



Yang, Eric. *Think Dinner. Mac Evolution, 1998*

Learn from Human Being's Brain

- Designing “**Accelerators**” to boost up **Machine Learning**



Domain Specific Architecture (DSAs)

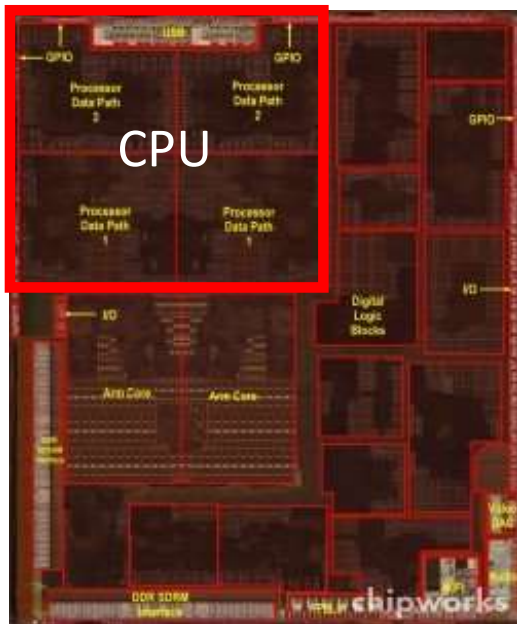
- Achieving higher performance by tailoring characteristics of domain applications to the architecture
 - Need domain-specific knowledge to work out good DSAs
 - Domain Specific Languages (DSLs) + DSAs (not strict ASIC)
 - Specialize to **a domain of many applications**
- Examples
 - GPU for computer 3D graphics, virtual reality
 - Neural processing unit (NPU) for machine learning
 - Visual processing unit (VPU) for image processing

Domain Specific Languages (DSL)

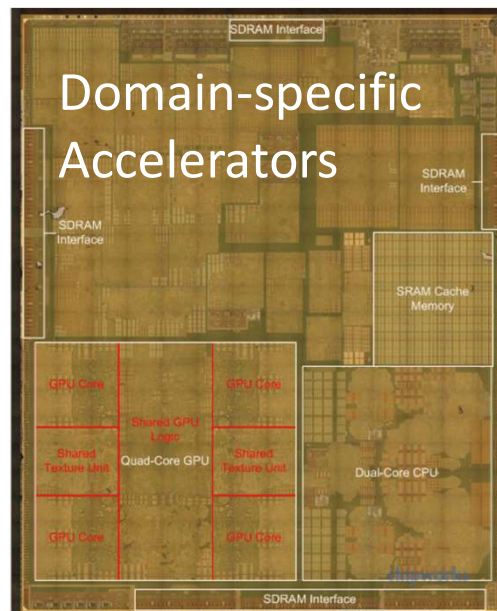
- DSLs target specific operations on a domain of applications
- Need vector, matrix or sparse matrix operations
- DSLs tailors for these operations
 - OpenGL, TensorFlow, Halide
- Compilers are important if DSLs are architecture-independent
 - Translate, schedule, map ISAs to right DSAs

Where is Domain-Specific Accelerators

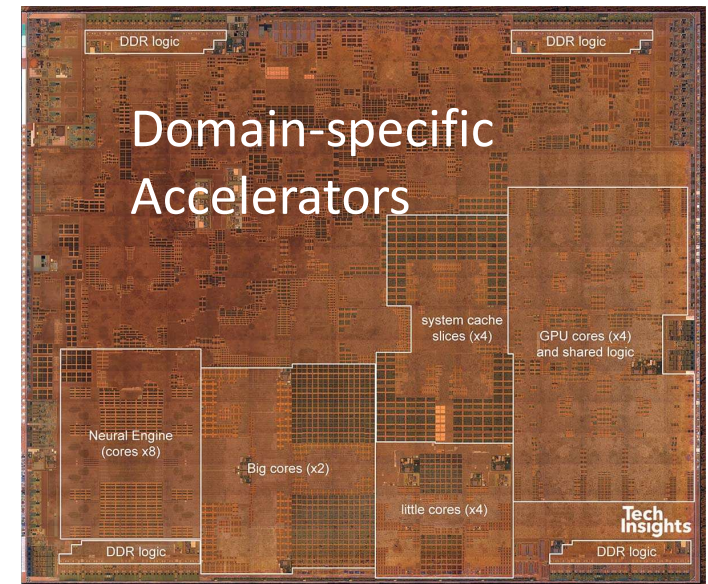
- Domain-Specific Accelerators are everywhere



2010 Apple A4
65 nm TSMC 53 mm²
4 accelerators



2014 Apple A8
20 nm TSMC 89 mm²
28 accelerators



2019 Apple A12
7 nm TSMC 83 mm²
42 accelerators

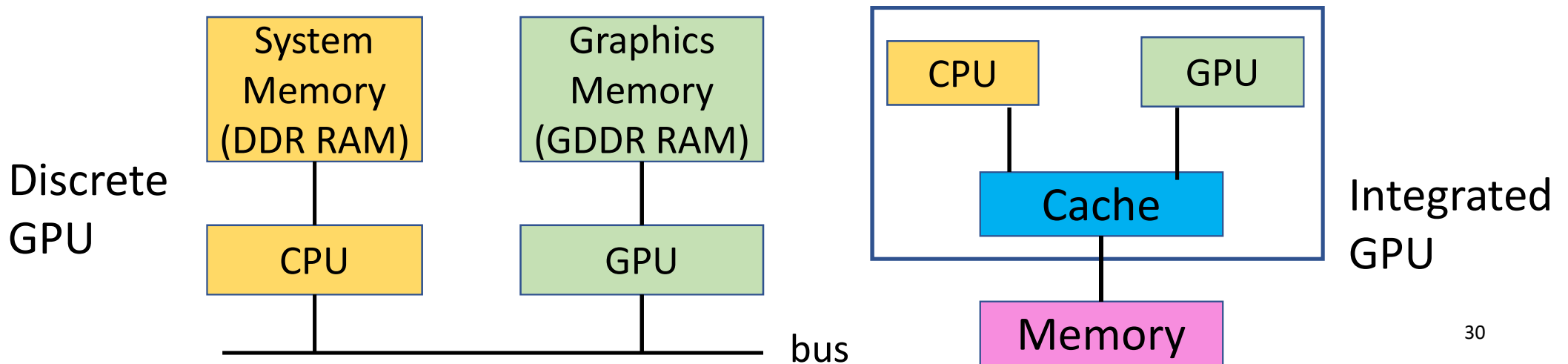
<https://edge.seas.harvard.edu/files/edge/files/alp.pdf>

Why DSAs can win ?

- More effective parallelism for a specific domain
 - SIMD vs. MIMD
 - VLIW vs. Speculative, out-of-order
- More effective use of memory bandwidth
 - User controlled vs. caches
- Eliminate unneeded accuracy (Quantization)
 - Lower FP/INT data precision (32 bit integers -> 8 bit integers)
- Increase the hardware utilization
 - Reduce the idle time on pipelining and LD/ST

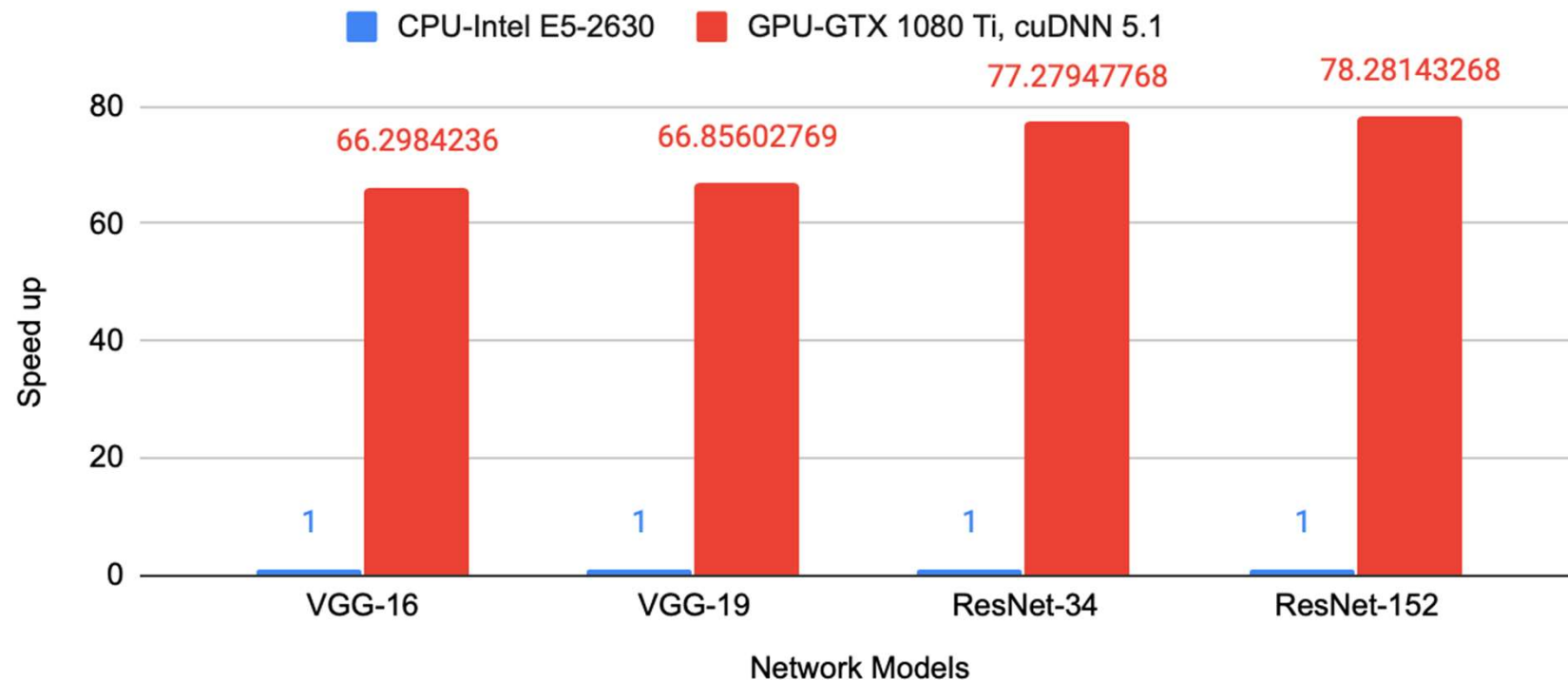
What is GPU?

- GPU = Graphics Processing Units
- Accelerate computer graphics rendering and rasterization
- Highly programmable (OpenGL, OpenCL, CUDA, HIP etc..)
- Why does GPU use GDDR memory?
 - DDR RAM -> low latency access, GDDR RAM -> high bandwidth




CPU vs GPU Training Time Comparison

- Normalized Training time on CPU and GPU (CPU has 16 cores, 32 threads)
- Why the model training on GPUs is much faster than on the CPU?



CPU vs GPU

	Cores	Clock Speed	Memory	Price	Throughput
CPU (Intel Core i7-7700k)	4	4.2 GHz	DDR4 RAM	\$385	~540 GFLOPs F32
GPU (Nvidia RTX 3090 Ti)	10496	1.7 GHz	DDR6 24 GB	\$1499	36 TFLOPs F32

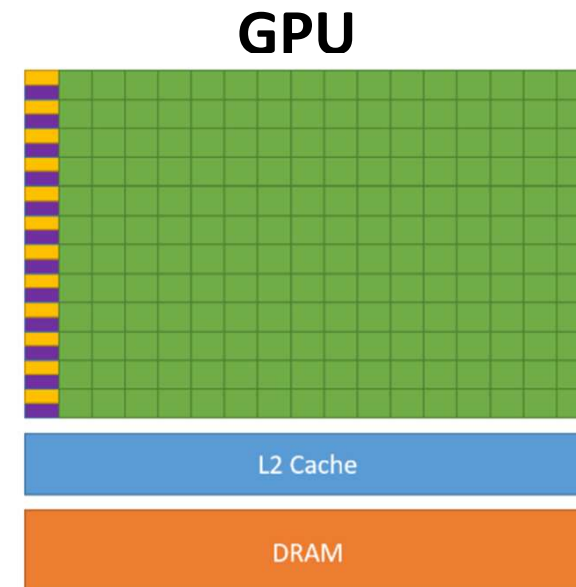
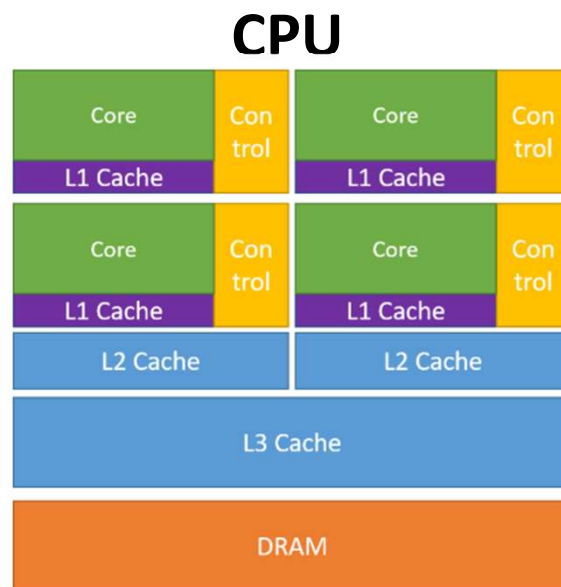


CPU: A **small** number of **complex** cores, the clock speed of each core is high, great for sequential tasks

GPU: A **large** number of **simple** cores, the clock speed of each core is low, great for parallel tasks

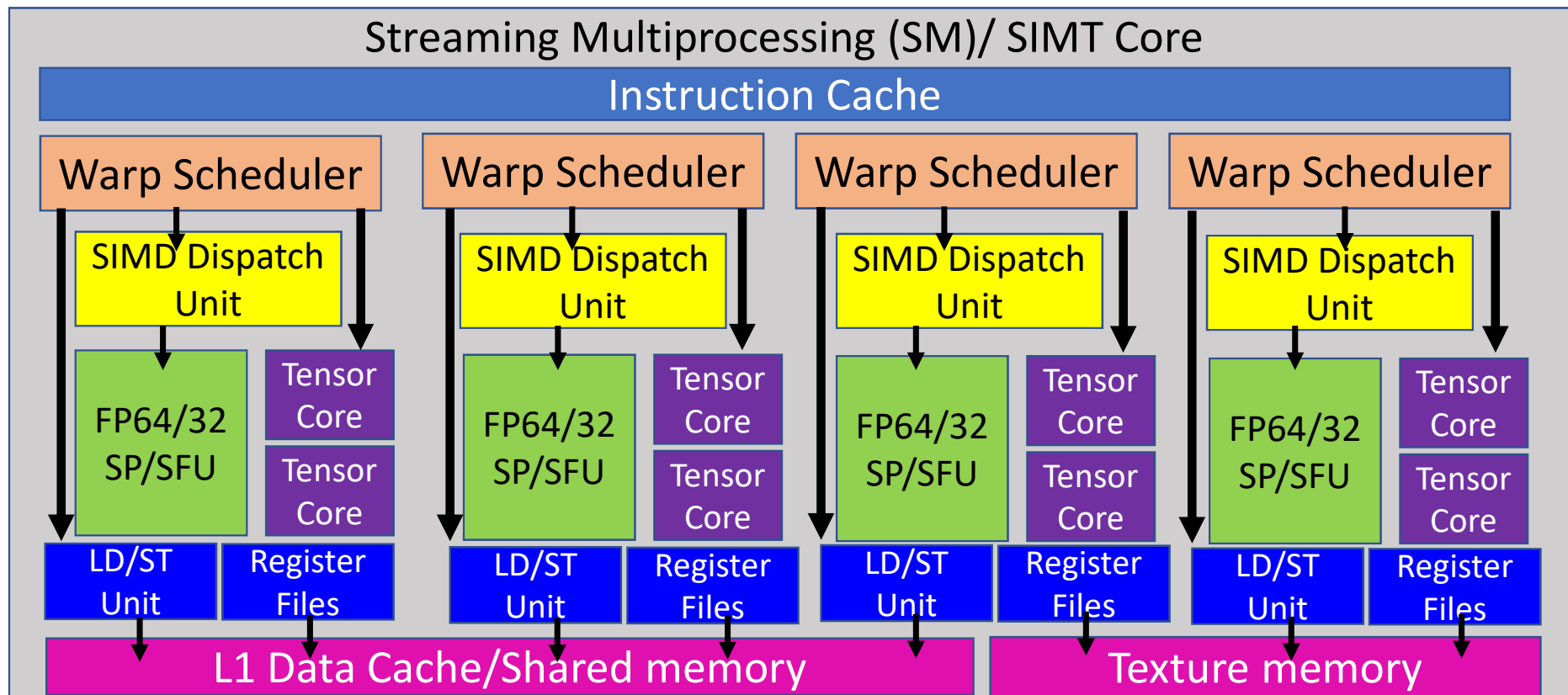
Why do we use GPU for computing ?

- What is difference between CPU and GPU?
 - GPU uses a large portion of silicon on the computation against CPU
 - GPU (2nJ/op) is more energy-efficient than CPU (200 pJ/op) at peak performance
 - Need to map applications on the GPU carefully (Programmers' duties)



What is Tensor Core on GPU?

- Execute $4 \times 4 \times 4$ matrix multiplication and addition in one cycle ($D = A \times B + C$)



Why do we need Tensor Core on GPUs ?

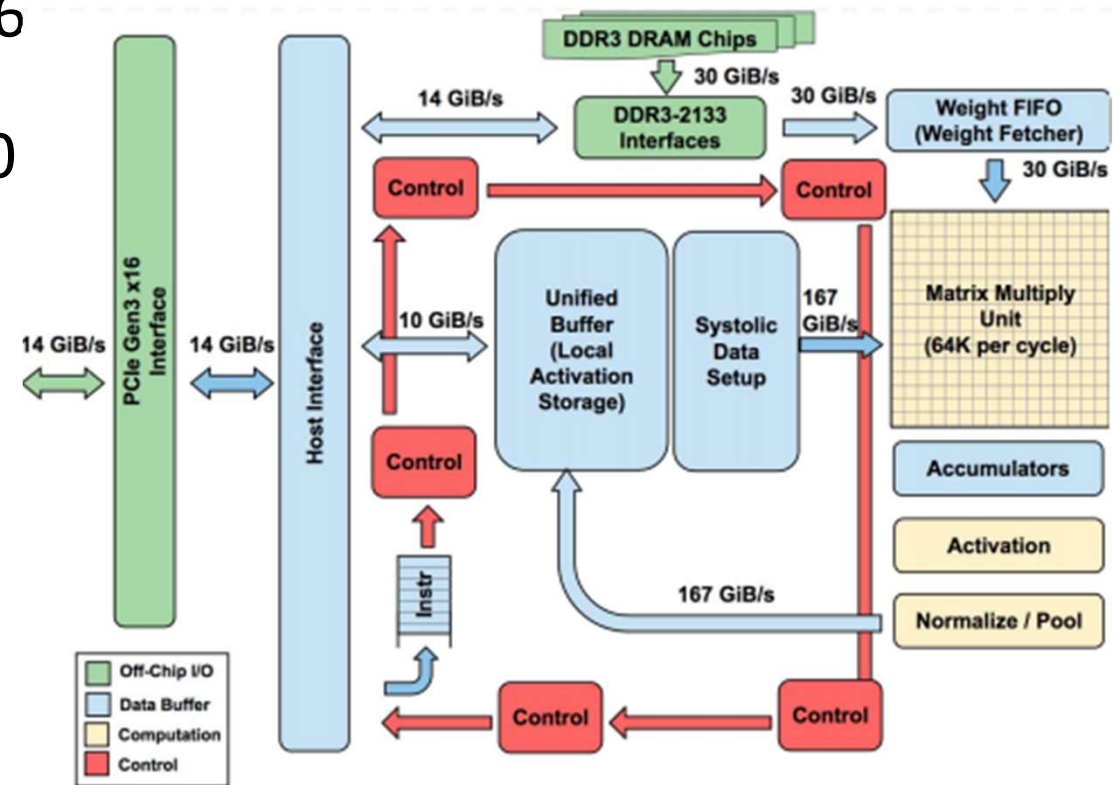
- Higher throughput for GEMM ?
 - A CUDA (SIMT) core offers 1 single precision multiply-and-accumulate operation per GPU cycle
 - Tensor core can multiply two 4 x 4 F16 matrices and add the multiplication product F32 matrix per GPU cycle
 - Tensor core can achieve **125 Tflops/s** vs **15.7 Tflops/s** for the single precision operation
 - Domain-specific Accelerator within the GPU

Story in Tensor Processing Unit (TPU)

- If people use DNN speech recognition service 3 mins per day
- Need to double Google's data center to meet this requirement
- Why not quickly a customized ASIC for inference ?
 - Need to **10 X** faster than GPUs
 - Must run existing apps developed for CPUs and GPUs
- Very short development time on TPU
 - Only take **15 months** for architecture and compiler invention, hardware design, build, test, deploy

Details in TPU v1

- **The Matrix Unit:** 64K (256 x 256) 8 bit INT multiply-accumulate
- Peak: 92T ops = 65536 x 2 x 700 MHz clock rate
- 4 MiB of 32-bit **Accumulator** collects 16 bit products
- Hardware **activation logics**
- 2.4 MiB **on-chip Unified Buffer** (Intermediate results)
- 3.5 X as much on-chip memory vs GPU
- 8 GiB **off-chip weight DRAM**



Performance Comparison

Processor	mm ²	Clock(MHz)	TDP (Watts)	Memory (GB/sec)	Peak TOPS/chip	
					8 b INT	32b FP
CPU: Haswell (18 core)	662	2300	145	51	2.6	1.3
GPU: Nvidia K80	561	560	150	160	--	2.8
TPU	<331	700	75	34	91.8	--

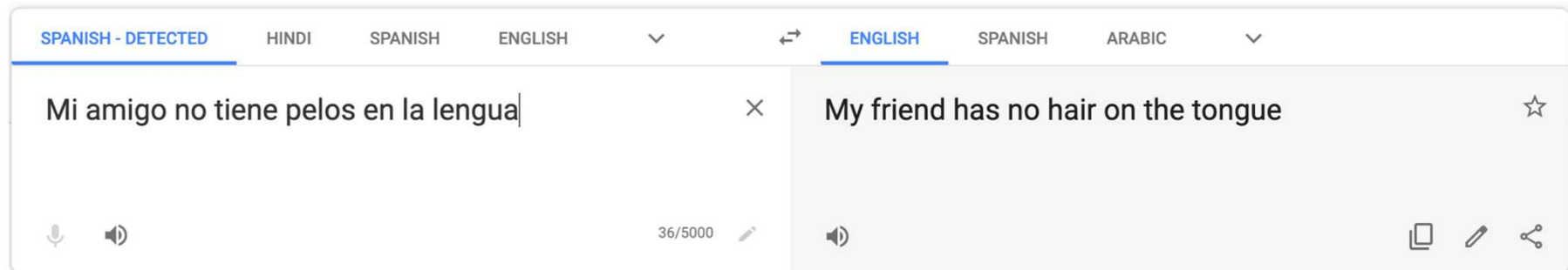
K80 and TPU in 28 nm process; Haswell fabbed in Intel 22nm process

Why TPU can Win ?

- Large matrix multiply unit
- Substantial software-controlled on-chip memory
- Data Quantization (8-bit INT)
- Parallelism on the hardware instead of Thread-level parallelism on GPUs
- What else ?

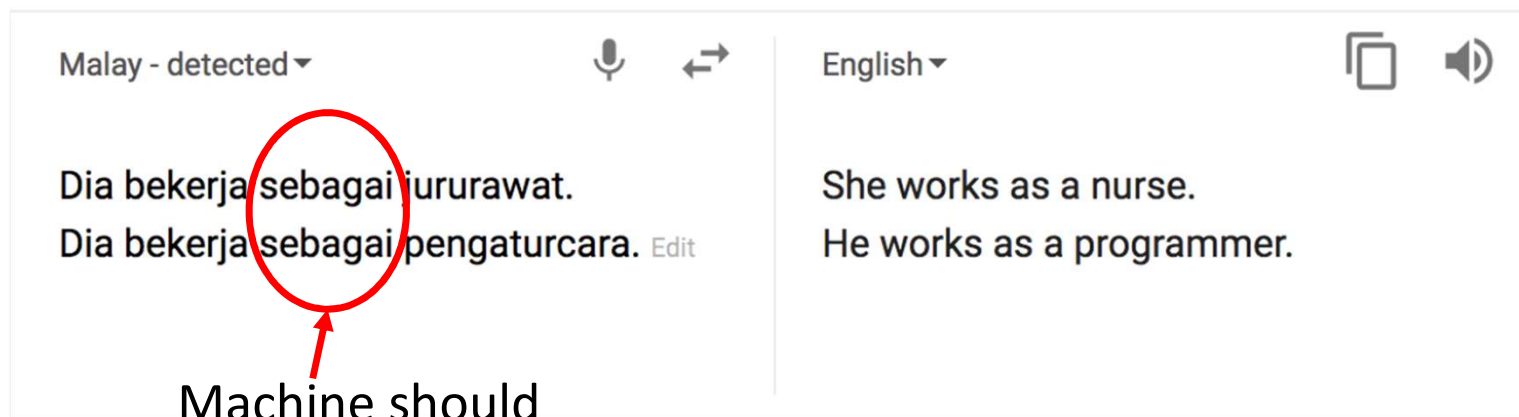
Is Machine Learning Perfect Enough ?

- **Nope !!**
- Many difficulties remain:
- Machine is still hard to figure out idiom
- Machine is also hard to understand common sense



Problems in DNN Language Translation

- **Nope !!**
- Machine has biases in the training data
- Why “**she**” -> nurse and “**he**”->programmer?

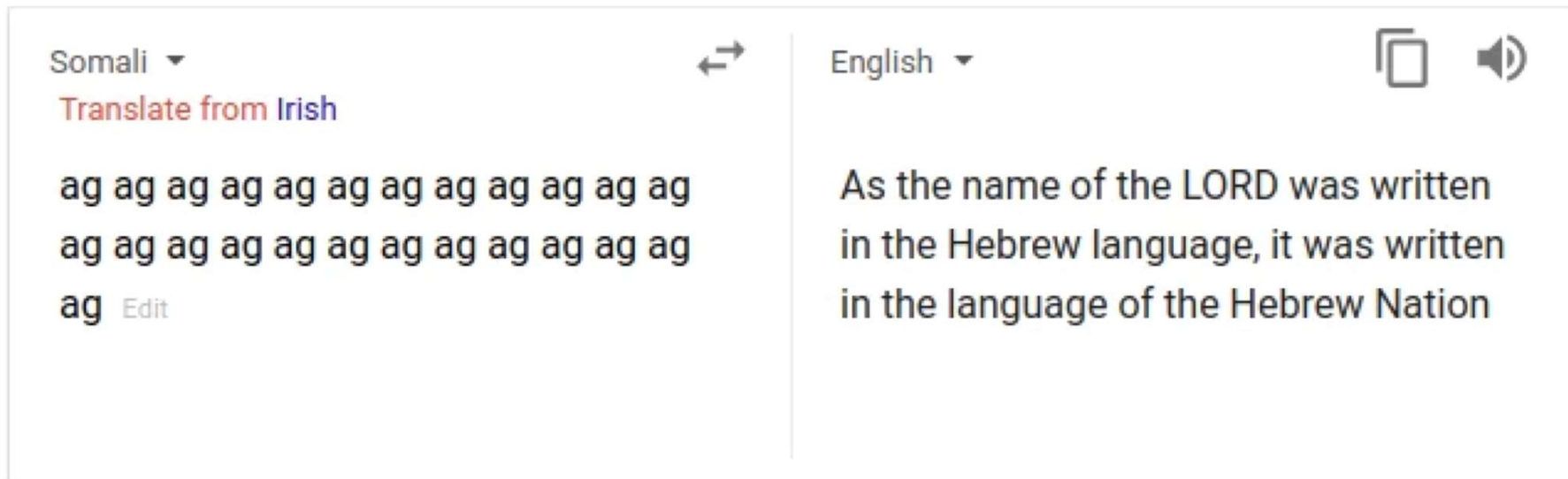


The screenshot shows a Google Translate interface with two columns. The left column is labeled 'Malay - detected' and contains the text: 'Dia bekerja sebagai jururawat.' and 'Dia bekerja sebagai pengaturcara. Edit'. The right column is labeled 'English' and contains the text: 'She works as a nurse.' and 'He works as a programmer.'. A red circle highlights the word 'sebagai' in the Malay text, with a red arrow pointing to it from the text below.

Machine should
specify the gender

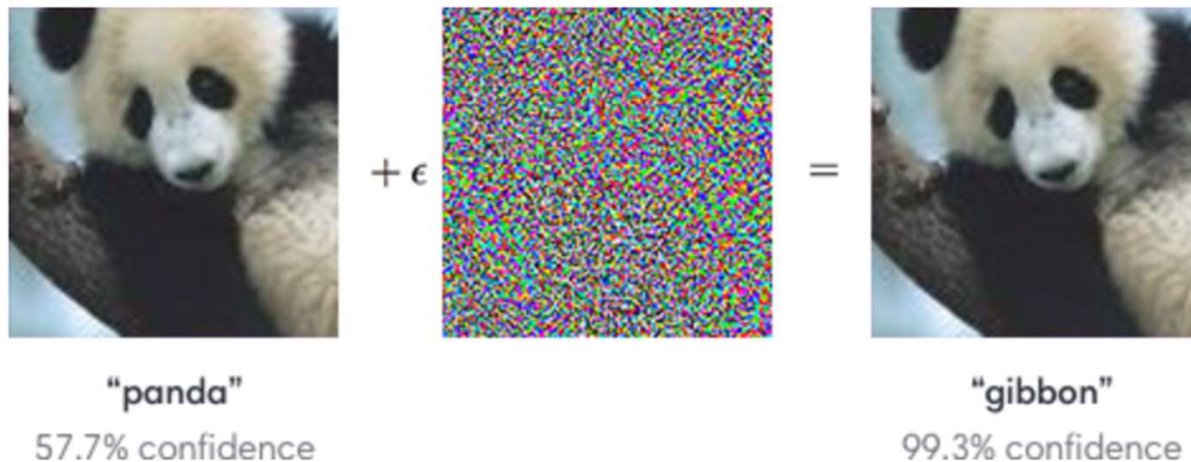
When Machine Learning System is Crazy

- Machine does something strange on uninterpretable system



Robustness of DNNs

- Adversarial Attack
 - Hackers add “noises” in your data (Adversarial samples)
 - Adversarial samples enable your DNN to be foolish
 - Reliability problems on self-driving vehicles using DNNs



<https://openai.com/blog/adversarial-example-research/>

My Research Work

Introducing Myself

- Lecturer: Tsung Tai Yeh
 - E-mail: ttyeh@cs.nctu.edu.tw
 - Office: EC 707
 - Research topics:
 - Computer architecture
 - Computer systems
 - Memory and storage systems
 - Domain-specific accelerators (GPU, Neural Processing Units)

“Hiring graduate and under-graduate students”

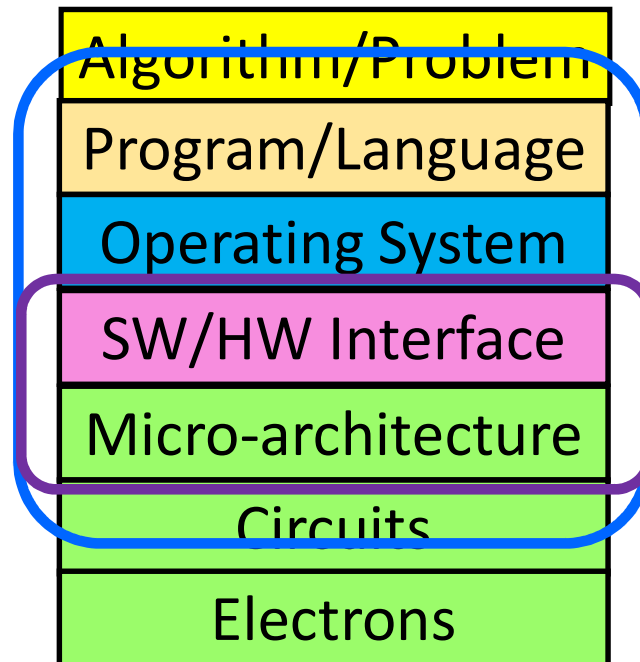


My Research Work

- To design the operational logics of computer devices
- Axiom: Improve “Energy Efficiency” and “Performance”

Computer Architecture (Expanded view)

Hardware-Software Co-Design (Algorithms to Devices)



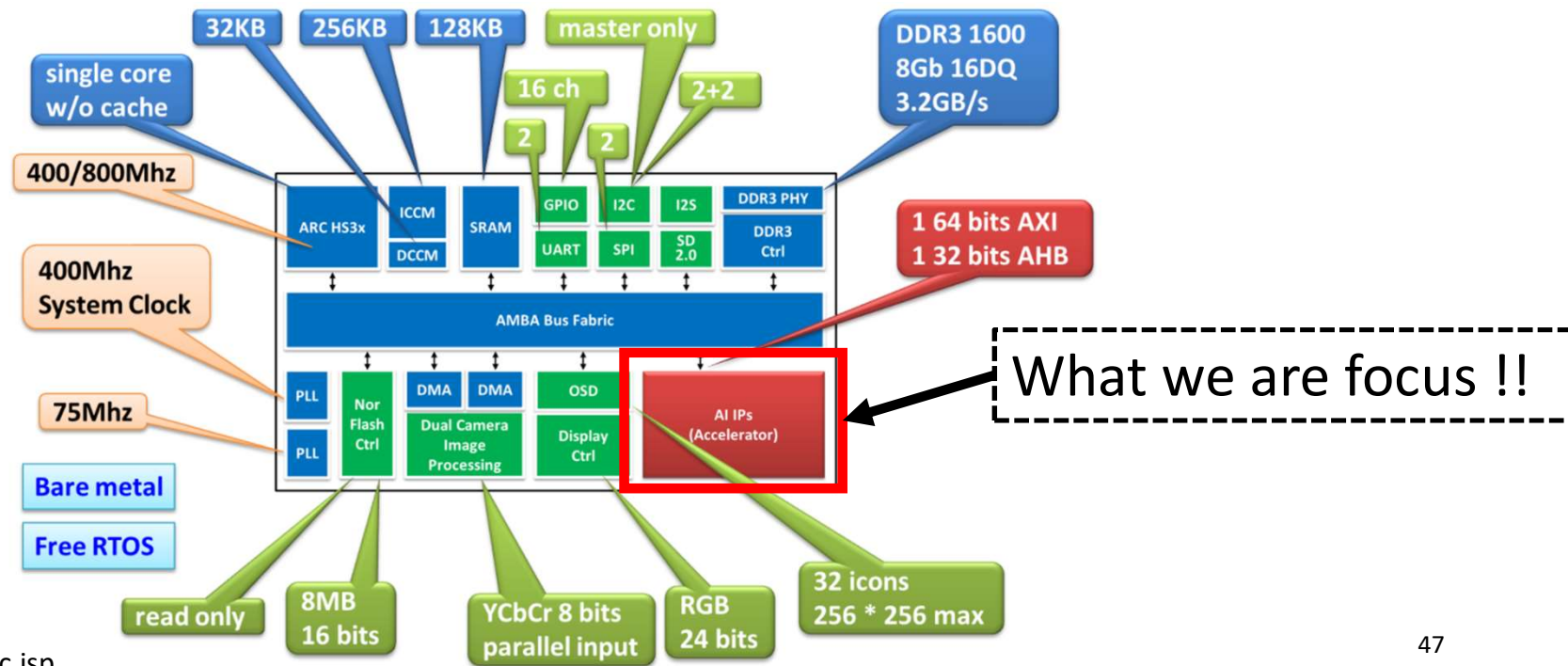
Computer Architecture (Narrow view)

Specialize on designs of SW/HW inference and Micro-architecture

Computer Architecture & System (CAS) Lab

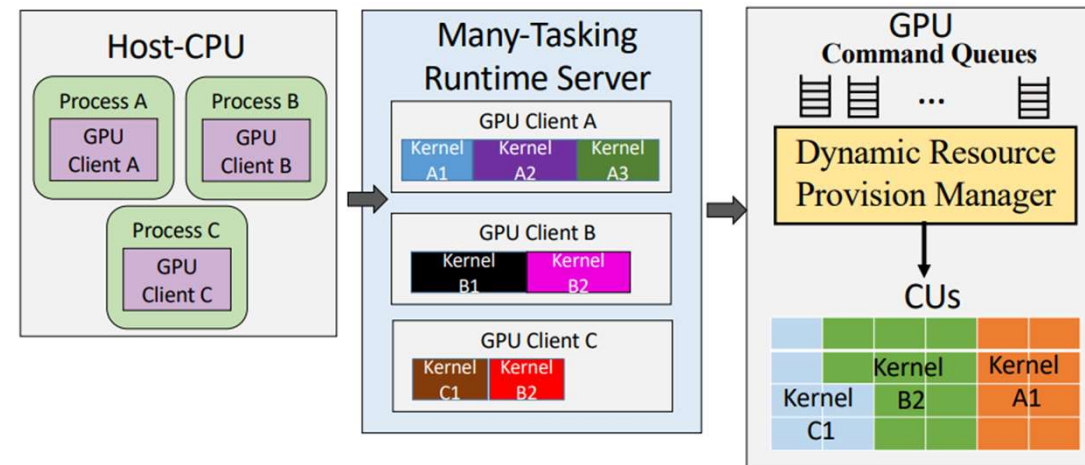
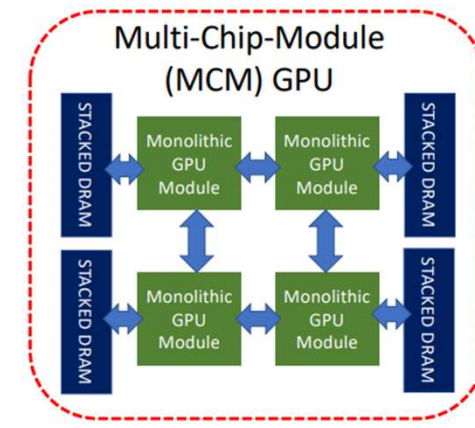
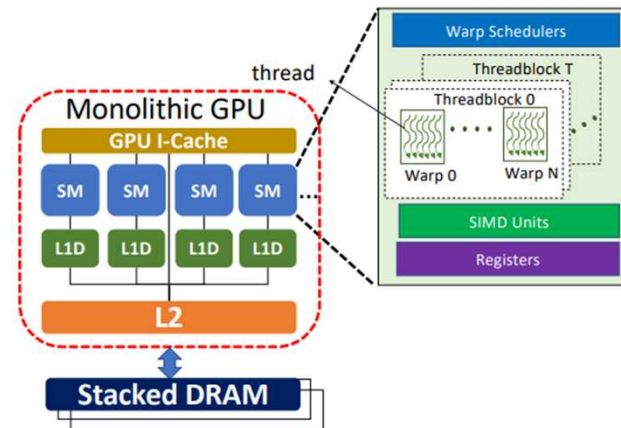
- Build better architectures and Systems
- Create hardware & software Intellectual Property (IP)

TSRI AI SoC
Design Platform
Block Diagram



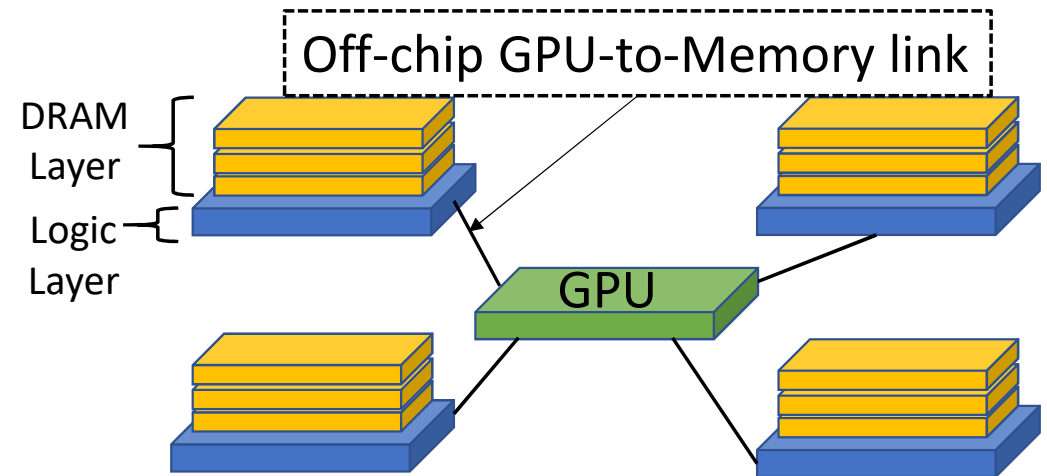
Multi-Tasking Computing

- Multi-tasking is everywhere
 - AI inference serving
 - Fintech (High Frequency Trading)
 - Networking/database
- Goals
 - High throughput
 - Low latency
 - High hardware resource util.
- Designs
 - QoS Scheduling
 - Virtualization



Low Power Edge GPU Architecture

- GPU acceleration on edge devices
 - Xbox, PS 5 -> Gaming
 - Video surveillance
- Goal
 - Low power
 - High memory bandwidth
- Design
 - New Nov-volatile memory integration

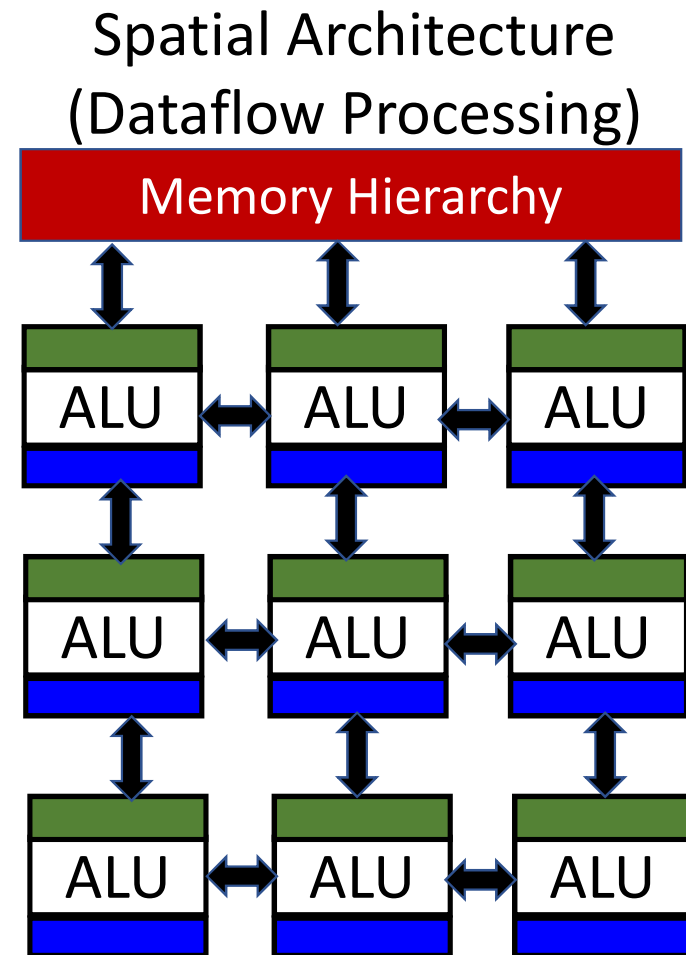


New Non-volatile Memory



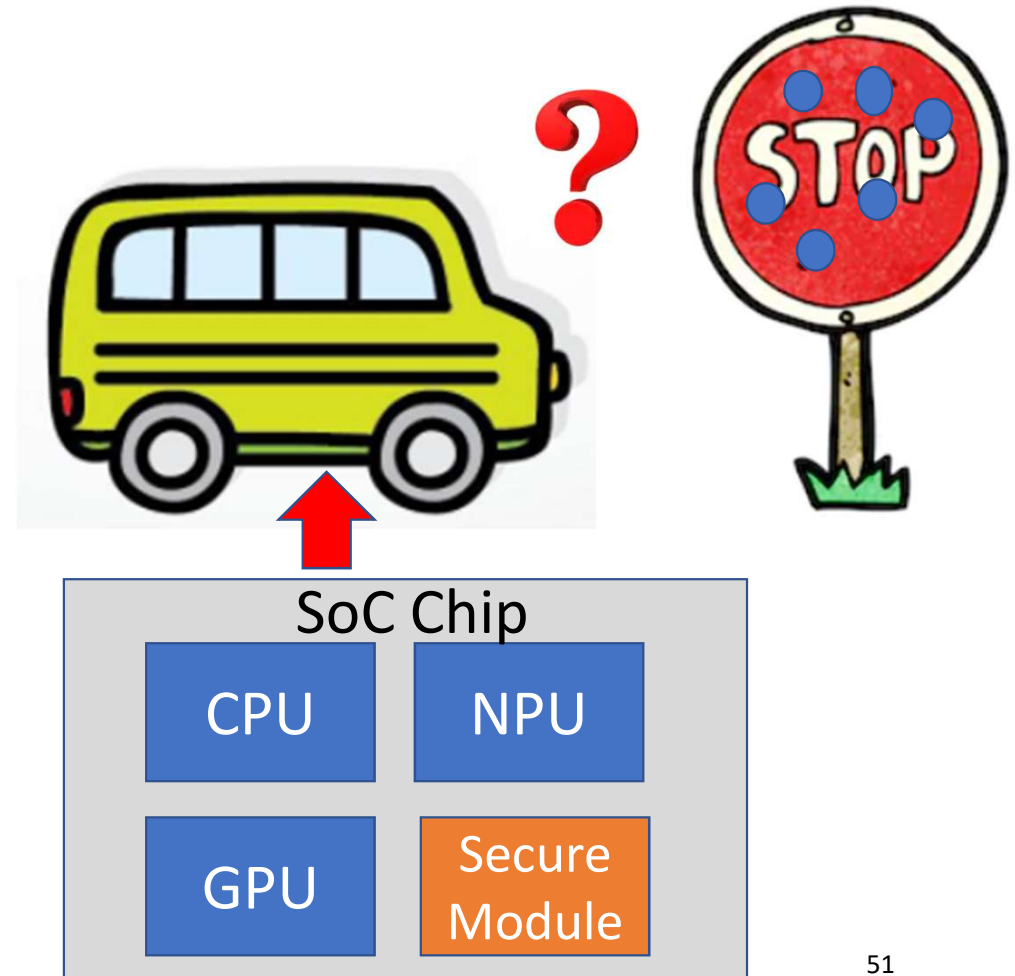
Dataflow DNN Accelerator

- Neural Processing Unit
 - Accelerate DNN models
- Goal
 - Energy Efficiency
 - Low Latency
- Design
 - Scheduling & Mapping
 - Network-on-Chip communication
 - Tiny ML



Robust DNN on Accelerators

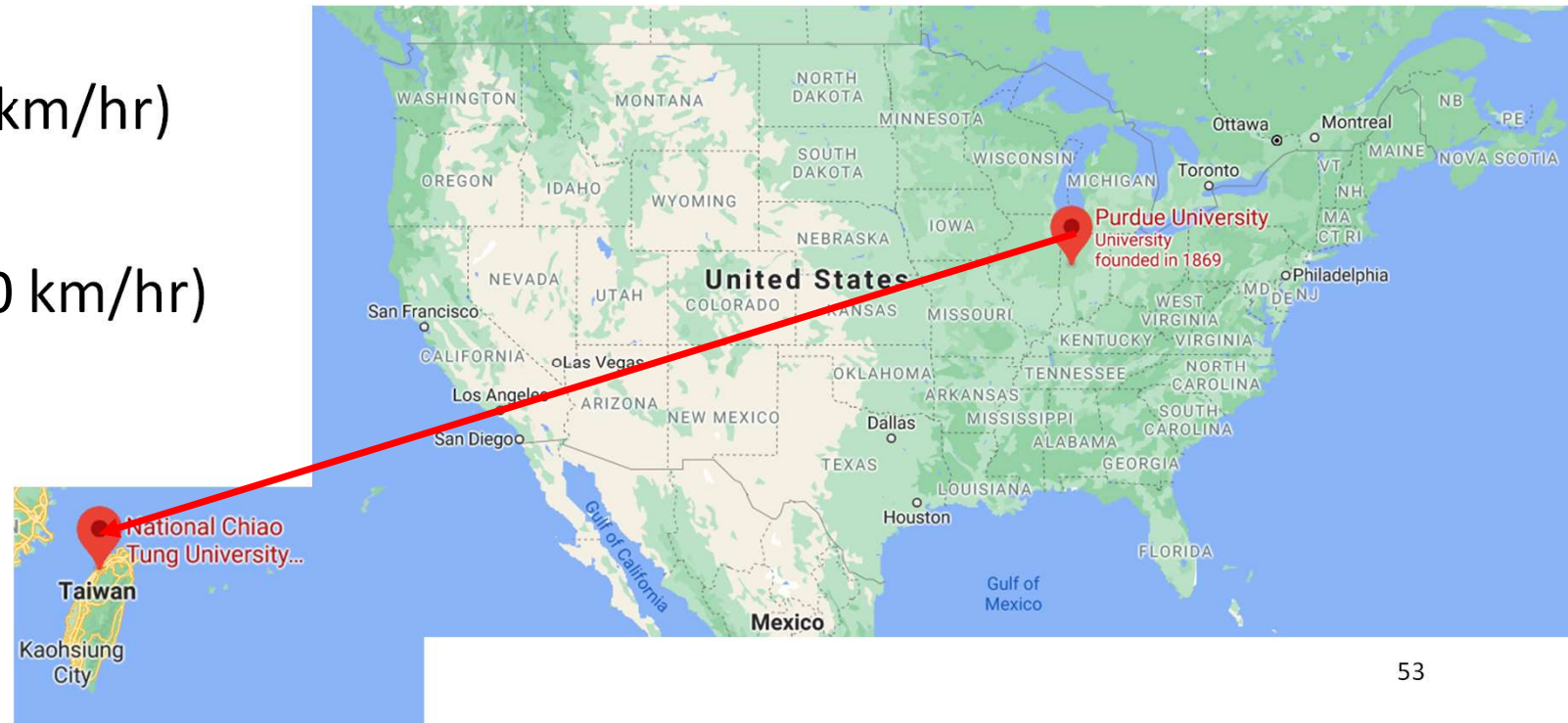
- If your self-driving car cannot recognize spotted stop sign ?
- Foolish DNN Models
 - Un-labelled training data
 - Malicious attack
- Self-driving vehicles have response time constraint
- Design
 - Defense algorithm
 - Accelerate secure defense



My Life @ U.S.

Where am I from ?

- How far from Purdue to NCTU? (Shortest Path)
 - 11777.55 km
- Taking Flight
 - 13 hrs (900 km/hr)
- Drive Car
 - ~5 days (100 km/hr)
- Walk
 - ~82 days (6 km/hr)



How does Purdue Look Like ?



Purdue Union

How does Purdue Look Like ?



Purdue alumni: Neil Armstrong

Seasons in Purdue ?



Beautiful Spring



Summer



Fall



Harsh Winter

Life @ Purdue

Coursework



Study



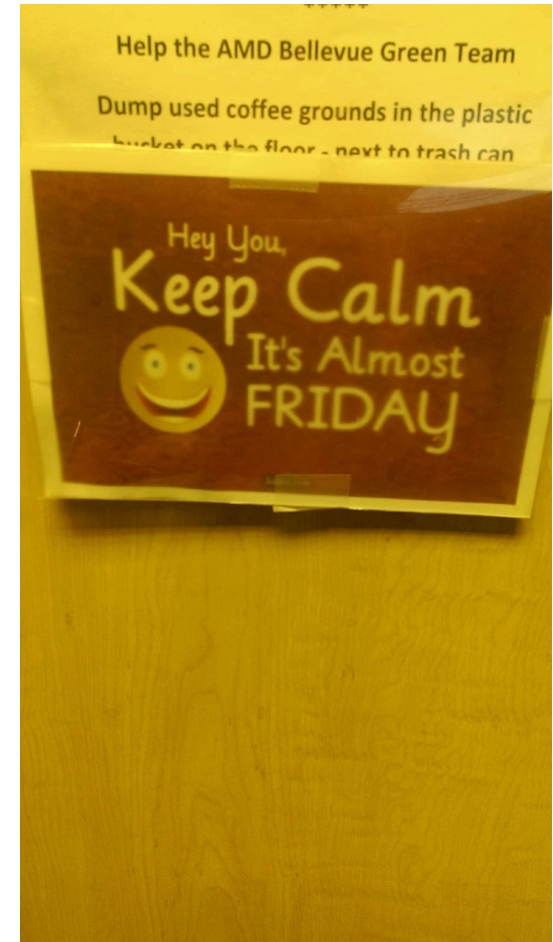
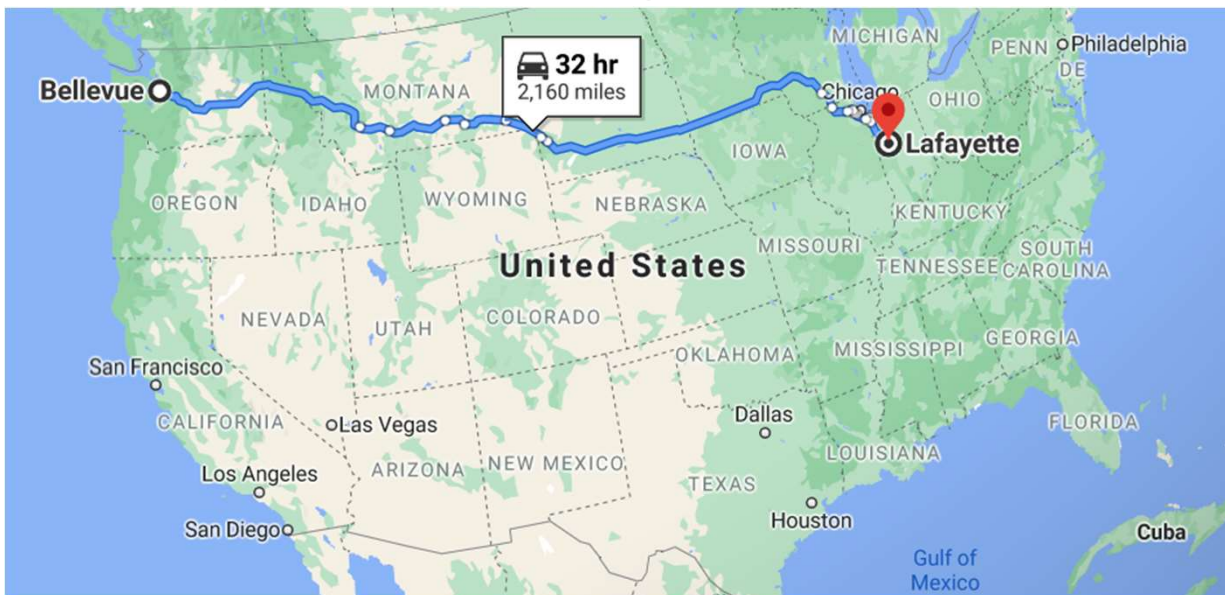
Sport



Trail

Internship @ AMD

- Work for ML on the GPU at AMD, Bellevue (6 months)
- Collaborate with talents around the world
- CompArch. Pub Night



Learn from the U.S.

- Work “**Hard**” and “**Smart**”
- Help yourself (Solving problems by yourself. Otherwise ...)
- Always ask “**Why**” when you don’t know
- “**Rejection**” is not the end of the life
- Enjoy the “**Beauty**” of the life



Words for Students

Advice for Freshman

- Not sure how many students read through my writing ?

Welcome to CS NCTU for class 2024 students Inbox x



Yeh, Tsung Tai <ttyeh@cs.nctu.edu.tw>

Fri, Sep 11, 8:37 AM

to bcc: a650993@gmail.com, bcc: danzel109.cs09@nctu.edu.tw, bcc: eric482695@gmail.com, bcc: ginny42222@gmail.com, bcc: huangweijie0310102@gmail.com, bcc: idexter.chuaa@gmail.com

Dear students,

Welcome to CS NCTU. You have worked so hard to get here, but your journey is just beginning. You all ace exams and might win championship games. However, the people of NCTU are some of the smartest people in Taiwan and will inevitably challenge you. Currently, you might be busy making new friends, exploring new places. I have some words for you before the beginning of this Fall semester.

1. To be a responsible person. You have grown up. You have become a college student, not a high school student anymore. Most people left home and released the leash from their parents. But, that does not mean you can be absent from your classes and do anything you like. Please behave like a grown-up person. Be responsible for your coursework, money spent, friends, and health. To have a regular life, work hard, have enough sleep. Your life will become simpler.
2. To learn to solve problems. Not every moment of college will be perfect and happy. You will encounter unexpected situations. You will have something you dislike. You will get tons of bugs when writing codes. Be confident and find a solution to overcome these challenges. Don't be afraid to reach out for help when you get stuck for a problem for a while. You are not alone.
3. To find out your passions. Try to take different classes in your **freshman** and sophomore years. Don't spend all of your time on particular courses. Do not use grades to measure the value of yourself. Try to explore what you like during your four-year college. Stick to your passions, do not limit yourself in your comfort zone. Dream big.
4. To learn to live with people. You can't do your college alone. Invest your time in people who want to know the real you. Show love to ones around you. All of you will be NCTU people, now and always.

Thank you for your patience if you have read through my words. Please seek out professors when you get confused with your coursework or college life. They will become your friends, your mentors, even your cheerleaders. I believe our memories and shared experiences at NCTU will be with us for a lifetime.

Your mentors,
Tsung Tai Yeh (葉宗泰)

Summary of my Advice

- Being a **responsible** person
- Learning to **solve** problems
- To find out your **passion**
- To learn to **live with people**



<https://www.storm.mg/article/2794353>

Conclusion

- The key of machine learning is “Learning”
- Smarter “machine learning” ?
- Need **Algorithm + Accelerator**
- **Sky is the limit**
- May you have a beautiful mind to explore the beautiful future



Thank You!!

Q & A